EDGE: An Enticing Deceptive-content GEnerator as Defensive Deception

Huanruo Li*, Yunfei Guo, Shumin Huo, and Yuehang Ding
National Digital Switching System Engineering and Technological Research Center
Zhengzhou, 450002, China
[e-mail: viaviavialhr@outlook.com]
*Corresponding author: Huanruo Li

Received January 27, 2021; revised March 13, 2021; accepted March 31, 2021; published May 31, 2021

Abstract

Cyber deception defense mitigates Advanced Persistent Threats (APTs) with deploying deceptive entities, such as the Honeyfile. The Honeyfile distracts attackers from valuable digital documents and attracts unauthorized access by deliberately exposing fake content. The effectiveness of distraction and trap lies in the enticement of fake content. However, existing studies on the Honeyfile focus less on this perspective. In this work, we seek to improve the enticement of fake text content through enhancing its readability, indistinguishability, and believability. Hence, an enticing deceptive-content generator, EDGE, is presented. The EDGE is constructed with three steps: extracting key concepts with a semantics-aware K-means clustering algorithm, searching for candidate deceptive concepts within the Word2Vec model, and generating deceptive text content under the Integrated Readability Index (IR). Furthermore, the readability and believability performance analyses are undertaken. The experimental results show that EDGE generates indistinguishable deceptive text content without decreasing readability. In all, EDGE proves effective to generate enticing deceptive text content as deception defense against APTs.

Keywords: Cyber deception defense, Decoy file, Fake text, Honeyfile, Honeypot

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0804004, and in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61521003.

http://doi.org/10.3837/tiis.2021.05.017 **ISSN : 1976-7277**

1. Introduction

As cyberattacks are getting increasingly sophisticated, digital information is confronted with rising threats. In particular, the Advanced Persistent Threats (APTs), a class of attacks conducted by highly organized attackers with low movement and usually steal data from the target system without being caught [1]. *Varonis's 2019 Data Breach Statistics* have stated that the largest APT has persisted for 30 years in Boeing, while only 3% of a company's folders are protected on average under investigation. Considering the insufficiency of perimeter-based security solutions, researchers design and deploy deceptive entities to actively defend and deter APTs without acquiring prior knowledge on attacks, that is, *cyber deception defense* [2][3]. Such defense mechanism not only collect, record, and report on underlying threats through deliberately exposing deceptive information, but also camouflages target assets.

The Honeyfile is a representative of cyber deception technologies. It is designed with deceptive content to distract attackers and attracts unauthorized use [4]. A typical Honeyfile system consists of monitor beacons, deceptive content, and deployment strategy [5]. The monitor beacons record unauthorized access and raise alerts [6][7][8][9]. The deployment strategy includes adaptive intrusion response and the interaction with the in-time monitor. Last but not least, deceptive content is a fundamental component to deceive and interact with attackers. The effectiveness of the Honeyfile system lies on the likelihood it attracts illegitimate access. Being the primary entity to interact with underlying intruders, the deceptive content has a direct impact on the effectiveness. Therefore, believable, indistinguishable (from neighbor digital context) and readable but inaccurate content will be more likely to deceive attackers and obfuscate their opinion to identify the fake from the real [10]. Our research improves the performance of decoy text content in enticing metrics including believability, indistinguishability, and readability.

The motivation of our work and major limitations of existing studies are: 1) Existing construction of decoy text is less effective to attract sophisticated attackers (i.e. APTS) due to the degradation on enticement. Decoy text used to contain random words and numbers [4][6], machine translations [11] and fake biographic information [12][13][14][15][16]. Cautious attackers will find it a trap and evades if they recognize the decoy content on previewing without access operation. 2) Studies on fake personal credentials, or formula and equations [8] are quite profound. However, less attention has been paid for technical context which is confronted with increasing risks. 3) Recent studies leverage novel techniques such as Natural Language Processing (NLP) [17] to generate decoy text. The readability is climbing through intelligent arrangement of fakes. However, as it takes partial original data as template to generate random context, the decoy text is presented with a different structure from local documents. The indistinguishability of decoy text is not fully considered.

Notwithstanding, creating decoy text content with NLP techniques and models fosters new opportunities to improve the enticement metrics, including readability, believability, and indistinguishability [18]. Hence, we propose a novel enticing deceptive-content generator, EDGE. Our aim is to improve the enticement of decoy text content for the Honeyfile to protect technical information. The ultimate goal is to escalate the probability to trap attackers and hide important technical information with deceptive information. Two main technical challenges need to be addressed in our research:1) How to extract important information in digital technical documents (defense target) and conceal them, 2) How to guarantee the enticement metrics of deceptive information.

For ***key information extraction***, existing studies extract important information mainly based on word frequency [8] [17]. However, words with high frequency may not fully cover the important information, especially for technical documents, where simple notations could convey specific message. In EDGE, we first remove stop-words and represents the rest of text content with N-dimensional word-vectors (denoted as concepts). We calculate the semantic similarity among concepts through semantic distance. Next, we leverage K-means++ algorithm to cluster all concepts. After rounds of iterations, EDGE obtains stable clusters and centroids. As centroids serve as the prototype of each cluster, the concepts at centroid could summarize the other words in the same cluster and host their semantic characteristics. Therefore, we extract these concepts as key information because they are more comprehensive to conclude the entire document. To replace such concepts, it will be more able to conceal original information and obfuscate attackers. To the best of our knowledge, we are the first to propose a combination of K-means++ algorithm with information extraction and summarization.

For ***the second challenge***, we guarantee the indistinguishability of decoy text through replacing important information with fakes to remain the structure of context. And the readability and believability of decoy text is realized by searching for deceptive candidate concepts with the Word2Vec model [19]. Because experiments show that the nearest neighbors of a word often reveal rare but relevant words that lie outside an average human's vocabulary in word-embedding model (i.e. Word2Vec). Integrating this, EDGE enables a better performance to replace important information with readable and believable decoy text than existing approaches using random words and numbers.

Last but not least, to strengthen the readability of decoy text, we include an Integrated Readability Index (IR) to supervise the final output of decoy text. This allows automatic update of enticement metrics of decoy text, which outperforms human-in-the-loop. The main contributions of this work are:

1) We propose a novel method, EDGE, to improve the enticement metrics of decoy text content for the Honeyfile by replacing semantic important information with readable and believable deceptive information;
2) EDGE addresses the decoy text generation and semantic important information extraction for technical documents, including patents and technical reports. In particular, a semantic distance-based K-means++ algorithm is presented to extract comprehensive semantic key information. (Section 4.2);
3) The construction of decoy text content leverages the Word2Vec model to search for uncommon but linguistically relevant words. This effectively strengthens readability of the decoy context but deepen obfuscation on attackers' comprehension. (Section 4.3);
4) Our proposal includes an Integrated Readability Index (IR) to supervise the final output of decoy text. For system with massive digital documents, automatic supervision of enticement metrics facilitates real-world implementation. (Section 4.4).

The rest of this paper is organized as follows: Section 2 reviews existing studies on the Honeyfile. Section 3 presents the threat model. The proposed methodology and implication examples are presented in Section 4. Section 5 presents experiment and evaluation results. Finally, Section 6 discusses the future work and draws a conclusion.

2. Related Work

2.1 Cyber Deception Defense

Deception has a rich history in the military attack and defense practices. Fred Cohen's paper [20] marked ***cyber deception*** as a promising defense solution in information security. Soon after this, the establishment of the Honeynet Project [4] extended the concept of cyber deception as "a digital or information system resource whose value lies in the unauthorized use of that resource". Nowadays, deception defense mechanism is known as" planned actions taken to mislead and /or confuse attackers and to thereby cause them to take (or not to take) specific actions that aid computer-security defenses" [21][22]. The key insight to mitigate APTs with cyber deception defense lies in cognitive manipulation. On one hand, APT attackers are supposed to be unaware of deception resources as they share high closeness to real assets. On the other hand, even if being conscious, attackers will have to spend more telling decoys before taking any action and might end up acquiring fake information [23][24].

Existing studies classify cyber deception defense techniques following their deployment in computer network systems [25] or interaction level with attackers [18]. For instance, Honeypot could be deployed in DMZ, Honeytoken in the runtime environment, and Honeyfile in application layer. As for the latter classification, techniques with higher interaction have deeper mimicry and greater realism to sustain a deception [26]. Low-interactive Honeyfiles mainly consist of random numbers and words while the high-interactive are built with realistic content [17].

2.2 The Honeyfile System and Deceptive Content

First proposed by Yuill et al. in 2004 [6], the Honeyfile system was implemented as an enhancement to an NFS and tested on the Honeynet. In their design, any file within the target NFS can be selected as a Honeyfile to send security alarms once being accessed. Later on, Bowen et al. [10] proposed an automatic decoy deployment system (the D^3 system), focusing on the automatic generation and deployment of decoy documents across the entire network system. Ben Salem et al. [5] conducted two user-studies finding that the use of decoy is effective and the placement of decoy files matters. In [7] [27], Voris et al. proposed the properties of decoys and studied the placement and combination of decoys in practice. Rowe et al. [28] have built a Fake Directory Generator, that generates deceptive Web interfaces with fake filenames pretending to be an authentic online directory. But fake files only contain either blocks of random words and numbers or error messages.

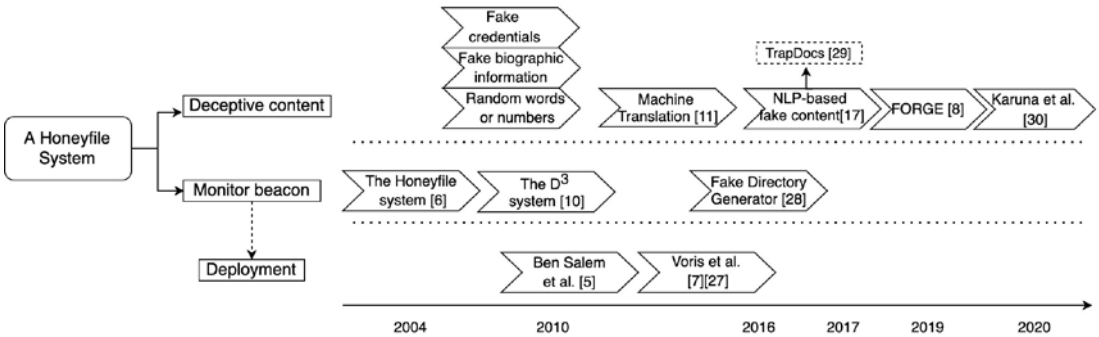


Fig. 1. A general structure and the development of the Honeyfile system.

A general structure and development of the Honeyfile system are concluded in **Fig. 1**. The study on monitor beacon is tightly associated with the deployment strategy to provide precise alarms and intrusion responses. From the perspective of deceptive content, previous techniques focused on either fake biographic information and identity credentials or random words under fake filenames. Nowadays, with the developing of attack tactics, attackers are getting more sophisticated, deceptive text content is in urgent need to be improved. Voris et al. [11] proposed to translated given text into different languages. Whitham [17] proposed a NLP based approach to generate high-interactive text content and avoid extra topics excluded in local directories. His design is later extended to, TrapDoc [29], an Honeyfile generator with an online version. Likewise, Chakraborty et al. proposed FORGE [8], an ontology based solution to generate deceptive formulas. Recently, the study of Karuna et al. [30] creates decoy text content through addition, deletion and mutation on text paragraphs. Deceptive modification operations are conducted by a genetic algorithm under comprehensibility and believability metrics proposed by the authors. The proposal generates comprehensible and believable fake text on reading exercises.

Our study mainly concentrates on generating deceptive text content, and the major limitations of existing approaches from this perspective are as follows:

- **Readability**: As the Honeyfile detects attackers only if it is accessed (read, write or move, etc.), the enticement in text content plays a significant role. Existing techniques [10][27][16][31] attempt to manipulate sensitive data or fake credentials (such as bank account, transfer link, PayPal account, and passwords) to increase its enticement. Others generate random words or select a random file [6] in the local directory. For sophisticated APT attackers who act mindful, those tricks are easy to be recognized [2]. Hence, Voris et al. propose [11] to obfuscate attackers' cognition via translating the given text into various languages. And [17] proposes to sort and rearrange words from the target directory in word frequency-based NLP manners to create high realism deceptive text content. However, neither of them addressed the readability of deceptive text content, a key issue to enticement [18]. In EDGE, a more effective posture in constructing readable deceptive text is proposed by semantic-sensitively extracting and replacing important information in given text content.
- **Automation**: From the perspective of feasibility, generating deceptive text content with hand-written rules [6][10][28] or human feedback [8] in the loop may lack scalability. EDGE automates the generation of deceptive content through integrating Readability Index in the election of optimal outputs.
- **Target scope**: According to Varonis report, intellectual property documents in the intranet environment are confronted with increasing threats. Studies on decoy personal information [12][13][14][15][16] and formulas or equations [8] are quite profound. However, decoy text for technical documents including technical reports and patents are insufficient. For instance, [30] proposes a novel approach to manipulate paragraphs to obfuscate attackers' understanding. But the concealment and manipulation of technical information is not addressed in this work as the train and test data are reading exercises. Hence, EDGE concentrates on generating enticing decoy text content for the Honeyfile system to protect and conceal significant technical information.
- **Deployable system and User evaluation**: Most existing studies have demonstrated deployable Honeyfile systems whether on monitor beacon or deceptive text content [6][5][10][8]. However, replication of the prototype system and the lack of universal quantitative metrics hinder the vertical analysis of existing solutions. In this paper, we propose to evaluate the effectiveness of EDGE with a qualitative believability experiment

and an Integrated Readability Index (*IR*) including Automatic Readability Index (ARI), Flesch Kincaid Grade Level, and Gunning Fog Index.

Table 1. A Comparison of EDGE with existing systems

	Honeyfile [6]	D^3 System [10]	Decoy Document [27][11]	Fake Directory Generator[28]	Whitham [17]	FORGE [8]	Karuna et al. [30]	EDGE
Readability	×	×	×	×	×	×	√	√
Automation	×	×	×	×	√	×	√	√
Target scope	Network file system	personal information	Business documents	Filenames	Technical documents (academic papers)	Formulas and equations	Reading exercises	technical documents
Deployable system	√	√	√	√	√	√	√	√
User evaluation	√	√	√	×	×	√	√	√

3. Threat Model

A large number of enterprises store, share and manage digital documents across intranet via remote access. Along with the convenience brought by the internet, the attack surface is expanding. For instance, APTs can penetrate into the intranet and hunt for valuable information [1]. APTs are vital threats to sensitive information assets but uneasy to catch because they are mindful in attack actions and persistent for attack goals [32]. Hence, researchers suggest mitigating APTs with deception defense that actively manipulates and obfuscates attackers' cognition on the target system.

In our threat model, we assume an APT attacker (from the Internet) has injected malicious applications on a legitimate host in a trusted intranet. With the malicious application, the attacker conducts stealthy remote access to the host without being detected. The attacker has the following capabilities to achieve the following goals:

- The attacker acquires primary access to digital documents in target intranet through remotely controlling the injected host in a trusted intranet.
- The attacker is capable of previewing parts of the documents without opening it. Normally, such attackers will not remove any file.
- The attacker aims to collect enterprise intelligence (such as important email copies, engineering design, and strategic plans, etc.) but he is unaware of the exact storing details.
- The attacker acts prudently to stay undetected for as long as possible and does not aim at exploiting vulnerabilities to compromise the entire system.

The defender's goal is to camouflage a group of important technical documents with the Honeyfile in the target intranet, and trap malicious access (from persistent attackers) with the deliberate leaked deceptive information. Generally, the Honeyfile is inaccessible to legitimate users. Only attackers enumerate files in system will touch decoy files [33]. The main purpose of a readable, believable and indistinguishable design of text content are: 1) increase the likelihood of being accessed, and 2) obfuscate attacker's recognition of the real information.

As we focus on deceptive content design in this work, we don't discuss the deployment strategy and monitor of Honeyfile, which are studied in [5][6][7][10][27].

4. Proposed Methodology

In this section, we present our methodology, EDGE, to generate enticing deceptive text content for digital technical documents. The fulfillment of key technical challenges: 1) How to extract important information in digital technical documents (defense target) and conceal them; 2) How to guarantee the enticement metrics of deceptive information; are thoroughly stated in the following.

4.1 Important Notations and the Pipeline of the EDGE

We regard the text content of a target digital document as a finite set of vocabularies, denoted as $T = C \cup S$, where S is a set of stop-words and the rest of vocabularies are a set of concepts, $\forall c_i \in C, C \subsetneq T, i \in N^*$. A specific subset will be denoted as *key concept* set if its concepts indicating important information, that is, $\exists c_j \in K, K \subseteq C, |K| = k < |C|$. For each key concept, l different candidate deceptive concepts will be searched and denoted as $c_j^l, l \in N^*$. To replace a set of key concepts with its candidate deceptive concepts, we obtain l copies deceptive text content $T'^l = (\sum_j c_j^l) \cup \complement_T K$.

Table 2. A summary of notations

Notations	Description
T	The text content of target digital document target directory
$T'^l = (\sum_j c_j^l) \cup \complement_T K$	Deceptive text content for T
S	A set of stop-words in T
$c_i \in C$	A set of concepts in T,$C \subsetneq T, i \in N^*$
$c_j \in K$	A subset of C denotes key concepts in T, $K \subseteq C, \|K\| = k < \|C\|$
c_j^l	Candidate deceptive concepts for $c_j \in K$, $l \in N^*$
$dist(c_m, c_n), c_m, c_n \in C$	Semantic distance between two concepts, $m \neq n, m, n \leq i$

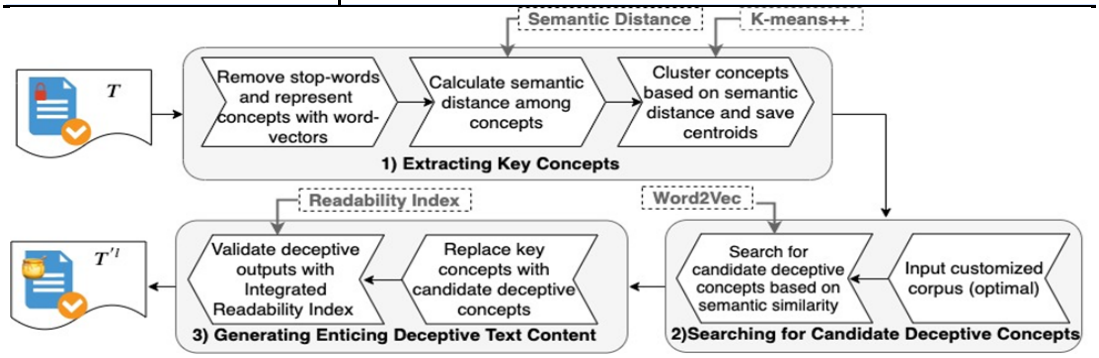


Fig. 2. The pipeline of EDGE.

The EDGE pipeline is sketched in **Fig. 2**, consisting of three steps:

1) *Extracting key concepts*: We initialize target digital documents with word-vectors and remove stop-words. To extract key concepts, EDGE has the following steps: First, we calculate the semantic distance between concepts represented by word-vectors. Afterwards, we cluster concepts under a semantic distance-modified K-means++ clustering algorithm. Cluster centroids are regarded as key concepts.
2) *Searching for candidate deceptive concepts*: For each key concept, EDGE searches for l different candidate deceptive concepts. To ensure the semantic similarity, we include a Word2Vec model [19].
3) *Generating deceptive text content*: In this step, we replace each key concept with sets of l candidate deceptive ***concepts***. We thereby obtain l copies of deceptive text content for the target digital document. Lastly, we validate the readability of T'^{l} with the Integrated Readability Index (IR). The most readable deceptive text content will be the final output.

4.2 Extracting Key Concepts

Key concepts extraction is the cornerstone of EDGE. In this step, we first demonstrate the definition of semantic distance. According to the hypothesis of Word2Vec model, where words can be denoted by vectors, semantic similar words tend to be assigned similar vectors. Hence, semantic distance could be calculated by L_2-distance between vectors and denote their semantic similarity.

Def 1: Semantic distance [34]

Let $dist(\boldsymbol{c_m}, \boldsymbol{c_n})$ denote the semantic distance, indicating the semantic similarity between $\boldsymbol{c_m}$ and $\boldsymbol{c_n}$. Each concept in text content T is denoted as $c_i \in C, C \subsetneq T, i \in N^*$, such that,

$$\boldsymbol{c_i} = f(c_i), \tag{1}$$

$$dist(\boldsymbol{c_m}, \boldsymbol{c_n}) = \| \boldsymbol{c_m}, \boldsymbol{c_n} \|_2 \tag{2}$$

Equation (1) represents the transformation of $c_i \in C$ into multi-dimensional word-vectors $\boldsymbol{c_i} \in \boldsymbol{C}$. In (2), $\| \boldsymbol{c_m}, \boldsymbol{c_n} \|_2$ denotes the L_2-norm of $\boldsymbol{c_m}$ and $\boldsymbol{c_n}$ that represents the semantic distance between them. Note that, $dist(\boldsymbol{c_m}, \boldsymbol{c_n}) \neq dist(\boldsymbol{c_n}, \boldsymbol{c_m})$ because word-vectors are directional. The smaller the distance is, the more likely that $\boldsymbol{c_m}$ and $\boldsymbol{c_n}$ share semantic characteristics [19].

As aforementioned, words with high frequency may not comprehensively cover the important information. Especially for technical documents, where simple notations could convey specific message. Therefore, we consider to maximize comprehension of extracted key concepts by proposing two constrains: 1) the semantic distance between key concepts and the rest should be as **small** as possible. This ensures that concepts are well summarized semantically by key concepts; 2) the distance between each key concept should be as **large** as possible. This prevents repeatedly selecting semantically similar words to replace while overlooking different important information.

Accordingly, we leverage the K-means++ algorithm to extract key information along with semantic distance. Seeing that the family of K-means algorithms are useful to partition n observations (vectors) into k clusters. With rounds of iterations, each cluster will belong to the nearest mean, that is, the cluster centroid. Moreover, in the K-means family, K-means++ has a better performance in initializing centroid, which accelerates convergence. That is, in K-means ++, n cluster centroids are randomly initialized and the $n + 1$ centroid is required to be as far as possible from previous n centroids in each iteration.

Combining K-means++ algorithm, words with similar meaning are grouped by clusters to fulfill the second constraint. For the first constrain, as the centroid serves as the prototype of a cluster, the centroid-concepts could summarize the other in the same cluster and host their

semantic characteristics. Hence, we take the centroid-concepts as key concepts, which comprehensively conclude the entire document. To replace such concepts will be more able to conceal original information and obfuscate attackers. To the best of our knowledge, we are the first to propose a combination of K-means++ algorithm with information extraction and summarization.

Lastly, the value of k, namely the limit of key-concept's amount, could be preset and adjusted as per security requirements. For instance, if the target digital document requires stronger deception, a larger k will be preset. The details are as follows:

1) Each concept in $c_i \in C$ is considered as a cluster node, $dist(\boldsymbol{c_m}, \boldsymbol{c_n})$ is the cluster distance between nodes. We use K-means++ to initialize cluster centroids with as large distance as possible;

2) $\exists c_i \in C, c_i \notin K$,$c_i$ will be clustered to its nearest centroids;

3) For each cluster, pick a node with the smallest sum in distance to the others as a new centroid and re-clustering;

4) Repeat 2) and 3) until centroids stay unchanged.

Algorithm 1 Extracting key concepts

Input: All concepts in T, $c_i \in C$

Output: Key concepts of C,$c_j \in K$, $K = centroids$

/* SEMANTIC DISTANCE CALCULATION*/

for $\boldsymbol{c_i}$ **in** C **do**

for $\boldsymbol{c_j}$ **in** K **do**

$dist(\boldsymbol{c_m}, \boldsymbol{c_n}) = \| \boldsymbol{c_m}, \boldsymbol{c_n} \|_2$;

/*CLUSTERING*/

$update\_centroids = ini\_(centriods(dist, n))$;

for c **in** $update\_centroids$ **do**

$cluster(\boldsymbol{c}) = \{\}$;

$cluster(\boldsymbol{c}) \leftarrow \{\boldsymbol{c}\}$;

while $centroids! = update\_entroids$

$update\_centroids = \{\}$

for $\boldsymbol{c}$ **in** C **do**

/*REPEAT UNTIL CENTROIDS STAY UNCHANGED*/

$$cluster\left(arg\min_{c_m \in K}\left(dist(\boldsymbol{c_m}, \boldsymbol{c})\right)\right).append(\boldsymbol{c})$$

for $\boldsymbol{c}$ **in** $centroids$ **do**

$$update\_centroids.append\left(arg\min_{\boldsymbol{c_m} \in cluster(\boldsymbol{c})}\left(\sum_{\boldsymbol{c_n} \in cluster(\boldsymbol{c})} dist(\boldsymbol{c_m}, \boldsymbol{c_n})\right)\right)$$

end

4.3 Searching for Candidate Deceptive Concepts to Replace Key Concepts

To guarantee the indistinguishability of decoy text, we propose to replace key concepts with fake information while remaining the structure of context. Our target scope is technical documents, patents and technical reports are included. These documents contain a large number of proper nouns. And the change of proper nouns will alter the meaning of original documents. Simply changing these words and important information with random words and numbers will decrease the believability and readability of decoy text.

EDGE leverages Word2Vec model, a word-embedding model, to search for candidate deceptive concepts. As experiments show that the nearest neighbors of a word (-vector) often

reveal rare but relevant words that lie outside an average human's vocabulary in the word-embedding model [35]. This enables a better performance to replace important information with readable and believable decoy text than existing approaches using random words and numbers.

The details of the designs are as follows. For each extracted key concept $c_j \in K$, EDGE will search for l candidate deceptive concepts c_j^l. The number of candidates, l, is customizable as per security requirements. Word2Vec model searches for semantic similar c_j^l *w.r.t* its extendibility and popularity. This model is popular and available to be trained with different corpus, which enables the EDGE to handle diverse text content. In searching for c_j^l, we exclude the plural forms of nouns, different tenses of verbs, adverbs of adjectives, vice versa, and capitalized forms.

Algorithm 2 Searching for candidate deceptive concepts

Input: Key concepts of C,$c_j \in K$;
The customized number of candidates $l = c, l \in N^*$
Output: Candidate deceptive concepts,c_j^l
/*TOP l SIMILAR CANDIDATE CONCEPTS*/
$count[a] = count[ad] = count[v] = count[n] = 0$
for c_j **in** K **do**
$c_i^l = most\_similar(c_j, topn = n)$
for $c_j == adj\&\&adv\&\&n\&\&v$ **do**
skip $cap.c_j\&\&ad.c_j\&\&a.c_j\&\&pl.c_j\&\&ad.c_j$
$count[i] += 1$
$i = [a, ad, n, v]$
count = $\sum_{a,ad,n,v}$count[i]
if $n - count = l$ **:**
break
end

4.4 Generating Enticing Deceptive Text Content

For each key concept, we have l candidate deceptive concepts. To replace $c_j \in K$ with c_j^l in T, respectively, we will have l copies of deceptive text content, $T'^l = (\sum_j c_j^l) \cup \complement_T K$. We validate the readability of T'^l with an Integrated Readability index and elect the most readable as the final outputs. An example presenting a pair of T and T'^l of US-41399506 when $k = 5, l = [1,5]$ is in **Fig. 3**. The Integrated Readability Index (IR) is a weighted sum of Automated Readability Index (ARI), Flesch-Kincaid Grade Level, and Gunning Fog Index. ARI is widely used on checking readability on all types of text, relying on a factor of characters per word rather than syllables per word. Flesch-Kincaid Grade Level is adopted by the U.S army to assess the difficulty of technical manuals and also used to score other legal documents including business policies and financial forms. And Gunning Fog Index estimates the years of formal education needed to understand the text on a first reading.[1] Such that, the Integrated Readability Index of T'^l is,

$$IR(T'^l) = \omega_a * R_A + \omega_f * R_F + \omega_g * R_G, \quad (3)$$

[1] https://en.wikipedia.org/wiki/Readability_test

$$\sum_{i=a,f,g} \omega_i = 1, \omega_i \in [0,1] \tag{4}$$

where the components of IR validate the readability of $T^{'l}$ from the perspectives of semantic readability, R_A, content complexity, R_F and technical specification, R_G. And the weights could be customized for different security needs. For example, some Honeyfiles are designed for technical documents requiring higher content complexity while others for sensitive business emails requiring better technical specification.
Such that, the output deceptive text content is,

$$\exists q \leq l, IR\left(T^{'q}\right) = max\left\{IR\left(T^{'l}\right)\right\}, q \in N^* \tag{5}$$

where $T^{'q} = (\sum_j c_j^q) \cup C_T K$. In the example, $\omega_a = 0.3$, $\omega_f = 0.35$, and $\omega_g = 0.35$.

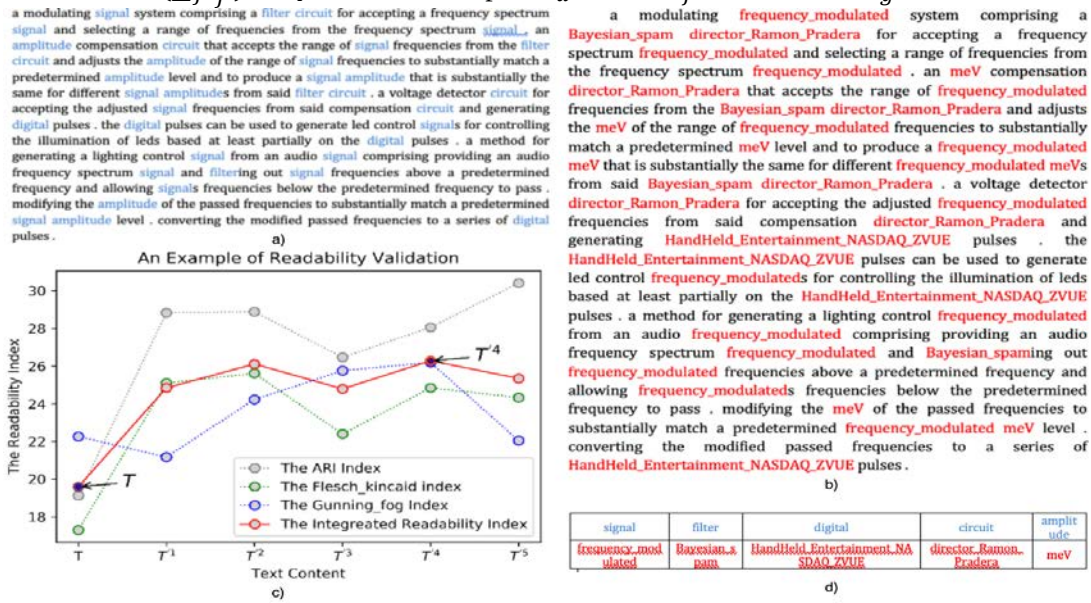


Fig. 3. The example of a pair of T and T'^{l}. a) is the target text content T,where key concepts are colored in blue, and b) is the deceptive text content $T'^{q}, q = 4$, where deceptive concepts are colored in red. c) plots the readability index of T and T'^{l}, and d) shows the mapping of key concepts to the deceptive when $k = 5, l = [1,5]$.

5. Experiment and Evaluation

In this section, we present our experiments to validate the proposed framework, EDGE. Experimental datasets and parameter setting are demonstrated in section 5.1. We conduct a readability performance analysis and a believability experiment.

5.1 Datasets and Parameter Setting

In our experiment, we extract two datasets. The first dataset with 50 digital documents harvested from the Circulation[2], an academic journal focuses on cardiovascular health and

[2] https://www.ahajournals.org/journal/circ

disease, is denoted as D_m. The other dataset with 100 US patent context extracted from the BigPantent [36], is denoted as D_p. For simplicity, each document in D_m contains extracted method and result from medical academic papers and each document in D_p is the abstract of the selected patents. To the best of our knowledge, existing studies in this field, excluding credential and biographic manipulation, mostly conduct experiments on academic papers [8][17]. We decide on the above datasets considering the target scope of our work for the following reasons: 1) we come after the precedent data collection principle to experiment on the text content of academic papers; 2) as it is very unlikely to test on real enterprises' intellectual property digital documents, we choose the medical academic method and results for they are plausible to share similarities with the context of development details and tests results in the medical industry; 3) the context of a patent is a common style of technical writing, which is supposed to mimic engineering design description text. Our datasets[3] are available for future study.

For believability experiment, D_m, D_p and their deceptive outputs will be distributed into four directories. For each dataset, target digital documents and deceptive outputs will be saved into test directories randomly and not repeatedly, but a pair of T and T'^q will not appear together. For all digital documents, we test under $k = 5$, 15, 30, respectively, and $l = 5$, that is, we conduct experiments on extracting 5,15, and 30 key concepts, respectively, and searching for 5 candidate deceptive concepts to validate outputs among 5 copies deceptive text content.

5.2 Readability Performance Analysis

Since one of the characteristics of the enticing deceptive concept is readable, we leverage quantitative readability metrics to analyze the performance of EDGE. A higher value of readability leads to greater realism and better enticement to attract attackers and sustain deception. Hence, we analyze with the Integrated Readability Index (***IR***) described in Section 4.4. Our benchmark is generated by the online version of TrapDocs. During the evaluation, deceptive text content for all digital documents in both datasets ($\boldsymbol{D_m}$ and $\boldsymbol{D_p}$), are generated under different preset size of $\boldsymbol{K}$, respectively. The integrated readability index is plotted in **Fig. 4**. The experimental outputs are denoted by $\boldsymbol{T'^l_{EDGE}}$ and the benchmark is $\boldsymbol{T'^l_{TrapDoc}}$.

It is shown in **Fig. 4** a) and b) that most T'^l_{EDGE} have higher Integrated Readability Index than T and $T'^l_{TrapDoc}$ take the lowest position for most of the time. We intuit that the higher IR is credit to EDGE replacing key concepts with semantic similar concepts and the automatic supervision to elect the most readable outputs. Also, comparing a) with b), we observe that the performance of both methods varies with target text content. Furthermore, in both cases, $IR(T'^l_{EDGE})$ is much closer to $IR(T)$ than $IR(T'^l_{TrapDoc})$. It could be concluded that T'^l_{EDGE} reads closer to T, that is, T'^l_{EDGE} is more indistinguishable in readability than $T'^l_{TrapDoc}$. We reason that TrapDocs degrades readability for its word frequency-based key concepts extraction and context rearrangement, while EDGE maintains context structure and manipulates on semantic key concepts. **Fig. 4** c) and d) illustrate the performance with extracting and replacing different amounts of key concepts. It could be observed that the increase of deceptive concepts, may slightly decrease the readability of T'^l_{EDGE}. But EDGE

[3] https://github.com/okkie-helloworld/EDGE_data

still performs well when a large portion of text content needs deceptive defense. In conclusion, we suggest that the EDGE is effective and steady to generate readable and indistinguishable enticing deceptive text content.

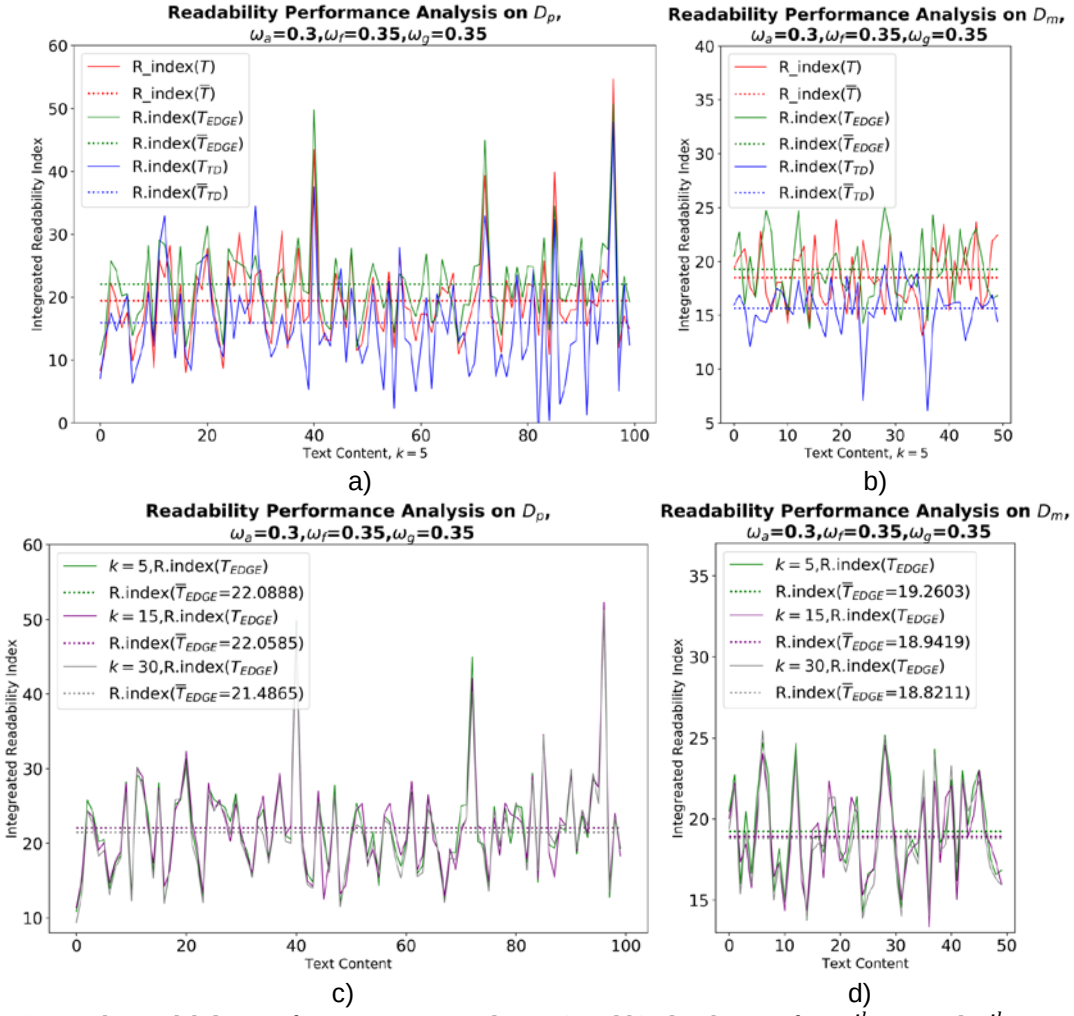


Fig. 4. The readability performance on D_p and D_m. a) and b) plot the IR of T, ${T'}^{l}{}_{EDGE}$ and ${T'}^{l}{}_{TrapDoc}$ for D_p and D_m, respectively. c)and d) plot the IR of all outputs when $k = [5,15,30]$.

5.3 Believability Experiment

The decoy believability experiment in [10] is conducted and includes a metric for a 'perfect decoy', that is,

$$Pr\left[Exp_{A,D,M}^{believe} = 1\right] = \frac{1}{2} \tag{6}$$

where D is a set of decoy documents in document space M, $D \subseteq M$. For any decoy denoted as $d, d \in D$, we have two documents $m_0, m_1 \in M(m_0 \neq m_1)$, such that either $m_0 = d$ or $m_1 = d$. For Adversary A attempting to identify an $m' \in \{m_0, m_1\}$ but $m' \neq d$ with a restrained cost (e.g. time, computing power). The output of the experiment is 1 if $m' \neq d$and 0 otherwise. $Pr\left[Exp_{A,D,M}^{believe} = 1\right]$ means that an adversary would choose to believe a decoy as a true file

with a probability of 1/2, indicating that a 'perfect decoy' is not 100% distinguishable from other normal files.

We evaluate the believability and indistinguishability of T'^{l}_{EDGE} following believability experiment. To understand medical academic paper, specific knowledge is required while the patent abstract is much easier for students with diverse background to acquire. Therefore, we model adversaries with intelligent cognition by engaging 20 students in medial and computer science, respectively, half of them are undergraduates and the other half are graduates. They are asked to review all documents in the preset test directories for D_m and D_p. Human subjects are asked to score 0 for deceptive files and 1 for authentic ones with their judgement, the results are illustrated in **Fig. 5**.

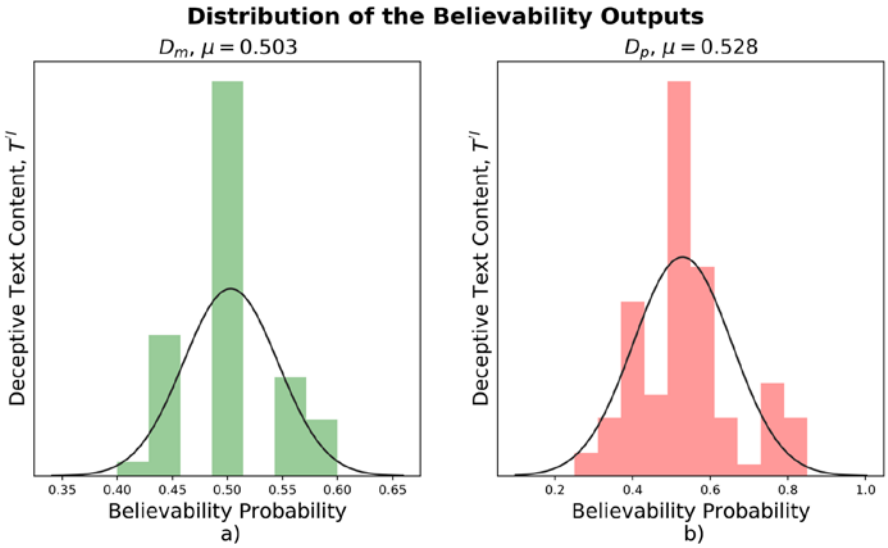


Fig. 5. Distribution of the believability experiment outputs. The y-axis represents the number of T'^{l}_{EDGE} drops in different believability probability and the black curve illustrates the normal distribution of outputs in both datasets.

In our experiment, human subjects score 0 when believing any as deceptive and 1 for authentic, such that the scores represent the probability of believability. With the average scores: 0.503 (**Fig. 5.** a)) and 0.528 (**Fig. 5.** b)) for each dataset, we thereby suggest that T'^{l}_{EDGE} could effectively confuse human cognition, that is, they are indistinguishable and believable deceptive text content.

6. Future Work and Conclusion

6.1 Future Work

According to Han et al. [25], a successful Honeyfile deployment includes enticing text content to sustain deception and a monitor system to observe and detect malicious behaviors. As EDGE addresses the first issue, a further study to improve monitor system and deployment is worthy of the future investigation. Considering the flexibility to manage applications through microservice architecture, we propose to develop Honeyfile into a cloud-based service [37]. Integrating monitor system and enticing text content generation into a microservice architecture will allow much more efficiency in both deceptive content generation and alarm

management. Moreover, our future studies will extend to the deployment strategy of the Honeyfile solution such as the distribution amount and locations.

6.2 Conclusion

Detection and defense against APTs are rather different from patching vulnerabilities due to their sophisticated orchestration and unpredictable movement. Cyber deception defense addresses this problem by proactively deceiving unexpected APT attackers with deploying baits or decoys in the target system. Honeyfile is a class of decoy documents that pretend to be target documents and mislead attackers to take action. High-interactive honeyfiles with greater enticement in text content will be more capable of sustaining deception with APT attackers.

In this paper, we propose EDGE: an enticing Deceptive-content GEnerator as Defensive Deception that focuses on improving the enticement of Honeyfile text content. EDGE is capable of creating enticing decoy text with minor decrease on the readability metric and effectively concealing important information. Also, we guarantee the indistinguishability of decoy text through keeping context structure to better attract stealthy APTs. To improve the performance on readability and believability, we propose a novel approach to comprehensively extract and replace important information. To the best of our knowledge, we are the first to propose semantics-aware key concept replacement-based deception text content generation. Lastly, we engage readability index in the pipeline to supervise the output, which enables automatic deployment in real-word environment. In the end, we analyze the performance of EDGE over a benchmark generator on datasets containing medical academic papers and patent abstracts. Also, we conduct a believability experiment to evaluate the believability and indistinguishability of EDGE. The results show that EDGE is steady to generate enticing text content for the Honeyfile system in scope of technical information with readability, believability, and indistinguishability.

References

[1] A. Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019. Article (CrossRef Link)

[2] N. Virvilis, B. Vanautgaerden, and O. S. Serrano, "Changing the game: The art of deceiving sophisticated attackers," in *Proc. of the Int. Conf. Cyber Conflict*, pp. 87–97, 2014. Article (CrossRef Link)

[3] S. Achleitner, T. La Porta, P. McDaniel, S. Sugrim, S. V. Krishnamurthy, and R. Chadha, "Cyber Deception," in *Proc. of the 8th ACM CCS International Workshop on Managing Insider Security Threats*, pp. 57–68, 2016. Article (CrossRef Link).

[4] L. Spitzner, "The Honeynet Project: trapping the hackers," in *Proc. of IEEE Symposium on Security and Privacy*, vol. 1, no. 2, pp. 15–23, 2003. Article (CrossRefLink)

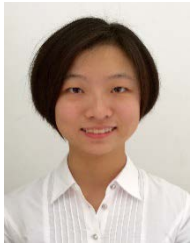
[5] M. Ben Salem and S. J. Stolfo, "Decoy document deployment for effective masquerade attack detection," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6739 LNCS, pp. 35–54, 2011. Article (CrossRef Link)

[6] J. Yuill, M. Zappe, D. Denning, and F. Feet, "Honeyfiles : Deceptive Files for Intrusion Detection," in *Proc. of the Fifth Annual IEEE SMC Information Assurance Workshop*, pp. 116–122, 2004. Artical(CrossRefLink)

[7] J. Voris, J. Jermyn, N. Boggs, and S. Stolfo, "Fox in the trap," in *Proc. of the Eighth European Workshop on System Security - EuroSec '15*, pp. 1–7, 2015. Article (CrossRef Link)

[8] T. Chakraborty, S. Jajodia, J. Katz, A. Picariello, G. Sperli, and V. S. Subrahmanian, "FORGE: A Fake Online Repository Generation Engine for Cyber Deception," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 2, pp. 518-533, 2021. Article (CrossRef Link)
[9] M. H. Almeshekah and E. H. Spafford, "Planning and Integrating Deception into Computer Security Defenses," in *Proc. of the 2014 workshop on New Security Paradigms Workshop - NSPW '14*, pp. 127–138, 2014. Article (CrossRef Link)
[10] B. M. Bowen, "Design and Analysis of Decoy Systems for Computer Security," Ph.D. dissertation, Dept. Comput. Sci., Columbia Univ., New York, NY, USA, 2011.
[11] J. Voris, N. Boggs, and S. J. Stolfo, "Lost in Translation: Improving Decoy Documents via Automated Translation," in *Proc. of IEEE Symposium on Security and Privacy*, pp.129–133, 2012. Article (CrossRef Link)
[12] E. De Cristofaro, A. Friedman, G. Jourjon, M. A. Kaafar, and M. Z. Shafiq, "Paying for likes? Understanding facebook like fraud using honeypots," in *Proc. of ACM SIGCOMM Internet Meas. Conf. IMC*, pp. 129–136, 2014. Article (CrossRef Link)
[13] S. Webb, J. Caverlee, and C. Pu, "Social honeypots: Making friends with a spammer near you,"in *Proc. of the 5th Conf. Email Anti-Spam, CEAS 2008*., pp. 1-10, 2008
[14] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. of 26th Annu. Comput. Secur. Appl. Conf. - ACSAC '10*, pp. 1-9, 2010. Article (CrossRef Link)
[15] N. Nikiforakis, M. Balduzzi, S. van Acker, W. Joosen, and D. Balzarotti, "Exposing the lack of privacy in file hosting services," in *Proc. of the 4th USENIX Work. Large-Scale Exploit. Emergent Threat. Botnets, spyware, Worms, More*, 2011.
[16] B. Liu, Z. Liu, J. Zhang, T. Wei, and W. Zou, "How many eyes are spying on your shared folders?," in *Proc. of ACM Conf. Comput. Commun. Secur.*, pp. 109–116, 2012. Article (CrossRef Link)
[17] B. Whitham, "Automating the Generation of Enticing Text Content for High-Interaction Honeyfiles," in *Proc. of the 50th Hawaii International Conference on System Sciences*, pp. 6069–6078, 2017. Article (CrossRef Link)
[18] B. Whitham, "Towards a set of metrics to guide the generation of fake computer file systems," in *Proc. of tthe 12th Australian Digital Forensics Conference*, 2014. Article (CrossRef Link)
[19] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv*, 2013. [Online]. Available: https://arxiv.org/pdf/1301.3781.pdf
[20] F. Cohen, "A note on the role of deception in information protection," *Comput. Secur.*, vol. 17, no. 6, pp. 483–506, 1998. Article (CrossRef Link)
[21] J. J. Yuill, "Defensive computer-security deception operations: Processes, principles and techniques," Ph.D. dissertation, Dept. Comput. Sci., North Carolina State Univ., Raleigh, NC, USA, 2006.
[22] M. H. Almeshekah, "Using deception to enhance security: A Taxonomy, Model, and Novel Uses," Ph.D. dissertation, Dept. Comput. Sci., Purdue Univ., West Lafayette, IN, USA, 2015.
[23] B. M. Bowen, P. Prabhu, V. P. Kemerlis, S. Sidiroglou, A. D. Keromytis, and S. J. Stolfo, "Botswindler: Tamper resistant injection of believable decoys in vm-based hosts for crimeware detection," in *Proc. of International Workshop on Recent Advances in Intrusion Detection*, pp. 118–137, 2010. Article (CrossRef Link)
[24] D. Fraunholz et al., "Demystifying Deception Technology:A Survey," *arXiv*, Apr. 2018. [Online]. Available: https://arxiv.org/pdf/1804.06196.pdf
[25] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Comput. Surv.*, vol. 51, no. 4, pp.1-36, 2018, Art. no. 80. Article (CrossRef Link).
[26] I. Mokube and M. Adams, "Honeypots: concepts, approaches, and challenges," in *Proc. of the 45th ACM Southeast Regional Conference*, pp. 321–326, 2007. Article (CrossRef Link)
[27] J. Voris, J. Jermyn, A. D. Keromytis, and S. J. Stolfo, "Bait and Snitch : Defending Computer Systems with Decoys," pp. 1–25, 2013. Article (CrossRef Link)
[28] N. C. Rowe and J. Rrushi, *Introduction to Cyberdeception*, Cham, Switzerland: Springer International Publishing, 2016, pp-1-8. Article (CrossRef Link)
[29] trapdocs. https://deception.ai/trapdocs/

[30] P. Karuna, H. Purohit, S. Jajodia, R. Ganesan, and O. Uzuner, "Fake Document Generation for Cyber Deception by Manipulating Text Comprehensibility," *IEEE Syst. J.*, vol. 15, no. 1, pp. 835-845, 2021. Article (CrossRef Link)
[31] M. Lazarov, J. Onaolapo, and G. Stringhini, "Honey Sheets: What Happens to Leaked Google Spreadsheets?," in *Proc. of 9th Workshop on Cyber Security Experimentation and Test (CSET 16)*, 2016.
[32] R. M. B. Secretary Acting, "Guide for Conducting Risk Assessments," 2011.
[33] J. Choi et al., "PhantomFS-v2: Dare You to Avoid This Trap," *IEEE Access*, vol. 8, pp. 198285–198300, 2020, Article (CrossRef Link)
[34] Y. Ding, H. Yu, J. Zhang, H. Li, and Y. Gu, "A Knowledge Representation Based User-Driven Ontology Summarization Method," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 9, pp. 1870–1873, 2019. Article (CrossRef Link)
[35] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014. Article (CrossRef Link)
[36] E. Sharma, C. Li, and L. Wang, "BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization," *arXiv*, 2019. [Online]. Available: https://arxiv.org/pdf/1906.03741.pdf
[37] A. Kyriakou and N. Sklavos, "Container-Based Honeypot Deployment for the Analysis of Malicious Activity," in *Proc. of 2018 Glob. Inf. Infrastruct. Netw. Symp.*, pp. 1-4, 2018. Article (CrossRef Link)



Huanruo LI was born in 1995, in China. She received the B.S. degree in Communication Engineering in 2017 and is now pursuing a PH.D. degree in Cyber Security from the National Digital Switching System Engineering and Technological Research Center (NDSC), Zhengzhou, Henan, China. Her research interests include cloud computing security and cyber deception defense.

Yunfei GUO was born in 1963, in China. He received the B.S. and M.S. degrees in Communication and Information System from the Beijing Institute of Technology, Beijing, China Since 2000, he has been a professor at National Digital Switching System Engineering and Technological Research Center (NDSC), Zhengzhou, Henan, China. He has authored three books, 18 patents, and more than 190 articles. His research interests include the next generation Internet, secure telecommunication, and cloud computing.

Shumin HUO was born in 1985. He received his M.S. and Ph.D. in Information and Communication Engineering from the National University of Defense Technology, China. He has been an assistant researcher at National Digital Switching System Engineering and Technological Research Center (NDSC), Zhengzhou, Henan, China. His research interests include theory and techniques of security for cyberspace and artificial intelligence.

Yuehang DING was born in 1995, in China. She received the B.S. degree in math from Harbin University of Science and Technology, Harbin, Heilongjiang, China in 2017. She is now pursuing a M.S. degree in Information and Communication from the National Digital Switching System Engineering and Technological Research Center (NDSC), Zhengzhou, Henan, China. Her research interests include knowledge graph and knowledge engineering.